International Journal of Science and Engineering Applications
Volume 2 Issue 7, 2013, ISSN-2319-7560 (Online)

FPGA Implementation of Real Time Data Acquisition System Using Micro blaze Processor.

D.Sathish kumar
CVR College of Engineering
EIE Department
Hyderabad, India

R.Ganesh
CVR College of Engineering
ECE Department
Hyderabad, India

Abstract— The Progression of the human existence from the primitive state to the present technological complex state is just an outcome of observation of this environment. By observing the environment and controlling the various physical parameters like temperature, pressure of the environment we are able to sustain on this beautiful earth. In order to control physical parameters there is a need of a specific system called Data Acquisition System (DAS) . Man has developed many different Data Acquisition Systems like Microprocessor based, PLC based, from Rock world to the Rocket world. This paper describes the implementation of Data Acquisition Systems using Embedded Processors.

Keywords — Data Acquisition, Virtex 5 FPGA Board, Micro blaze, GPIO, RTD Transducer

1. INTRODUCTION

In the real world all the physical quantities are continuous and measurable in analog domain, whereas in the electronic world of microprocessors, data storage, and automation is a digital domain. The conversion of parameters from analog domain to digital domain is the major advancement in semiconductor world with improvement in sensitivity, speed of conversion and parallel processing.

Data acquisition is the process of measuring quantities in the analog domain and converting it into the digital realm where any digital device can understand, record, and process and provides corresponding output in analog domain. The output could be visual, audible or any other electrical signal. The traditional method of data acquisition was to access analog data from field level and generates corresponding control action using comparators or any other analog circuitry.

This paper describes a way to design and implement a high performance and parameterized DAS on a single Virtex5 FPGA, which has more flexibility, power-efficiency, reconfigurabilty. The proposed DAS design can be implemented by using Xilinx Embedded Development Kit (EDK) ver. 12.4.

2. DATA ACQUISITION SYSTEM

Data acquisition products serve as a focal point in a system, tying together a wide variety of products, such as sensors that indicate temperature, flow, level, or pressure.

A digital data acquisition system includes the blocks shown in Figure.1.The essential operations of data acquisition system are:

1. Handling analog signals
2. Measurement
3. Data conversion
4. Programming and control

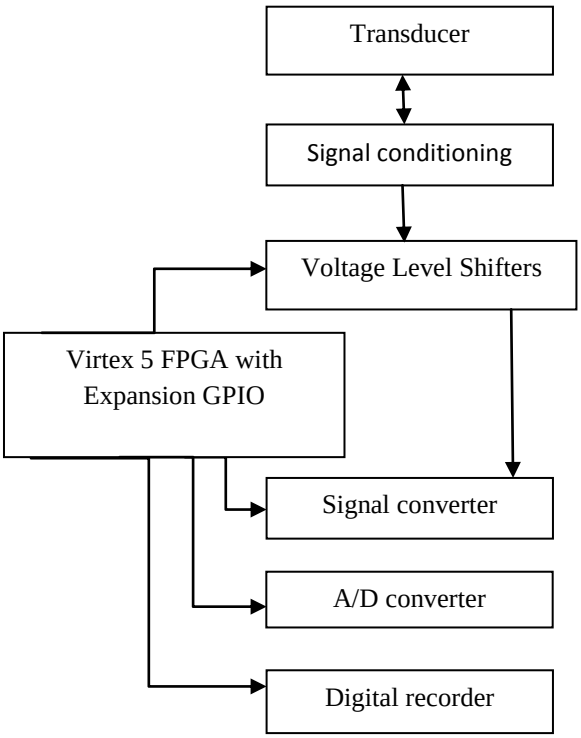

Figure 1. Block diagram of FPGA - DAS

The process of data acquisition is a step by step process, starting from sensing or collecting stimuli from external source or primary sensor to the final stage of proper alignment and storage of data in specific time duration.

3. COMPONENTS OF DATA ACQUISTION

The following are the different blocks or components of Data acquisition system:

1. Transducers
2. Signal conditioning circuits
3. Voltage translators
4. Analog to Digital converter
5. Microblaze
6. GPIO
7. Digital recorders

3.1. Transducers

They convert a physical quantity into an electrical signal which is acceptable by data acquisition system.

For eg: Platinum RTD which functions on based on positive temperature coefficient principle can be used to acquire the temperature.

$R_T = R_o [1+\alpha_o(T - T_o)]$; where Rt is the RTD resistance for change in temperature.

3.2.Signal conditioning circuit

The signal conditioning is essential as the transducer electrical output may be in micro or milli volts. Proper signal conditioning circuitry can be selected depending on transducer. Fr eg RTD can be connected to a Wheatstone bridge, Capacitive Transducer can be connected to a OP Amp circuit having a rich gain factor (β). The strengthened voltage can be connected to next phase of DAS.

3.3. Voltage Translator

The toughest part of DAS is configuring the voltage levels of the FPGA with Real time devices. The FPGA contains expansion headers on board, which are called as GPIO slots. (In addition GPIO soft IP is also presented in EDK tool.) Through these Expansion headers the real time devices like ADC; DAC can be connected in order to acquire physical data by the Microblaze processor of FPGA. Hence there will be a requirement of voltage adjustments between FPGA and outside ADC chip. Virtex 5FPGA supports 3.3 V signals, ADC delivers 5V signals. Appropriate signal conditioning level shifters have to be selected to overcome this typical phase of DAS Implementation.

3.2. Analog to Digital Converter

An A/D converter e.g.: ADC0809 is a data acquisition - successive approximate component is a monolithic CMOS device with an 8 bit digital out, 8 channel multiplexer, and microprocessor compatible logic. The working of ADC can be programmed by the timing signal study where its initiation and control signals have to program by GPIO of Micro blaze in Virtex 5.

3.3.Micro blaze

The Micro blaze is a soft processor core designed for Xilinx-FPGAs from Xilinx. The Embedded development kit [5], [6] is used to implement hardware design consisting IP cores and a Microblaze Soft Processor. The DAS design is implemented by writing the ADC logic in EDK software application to run on the Microblaze processor. The software application controls the functionality of different IP cores added to the processor. SystemC [4] programming language is used for developing software application

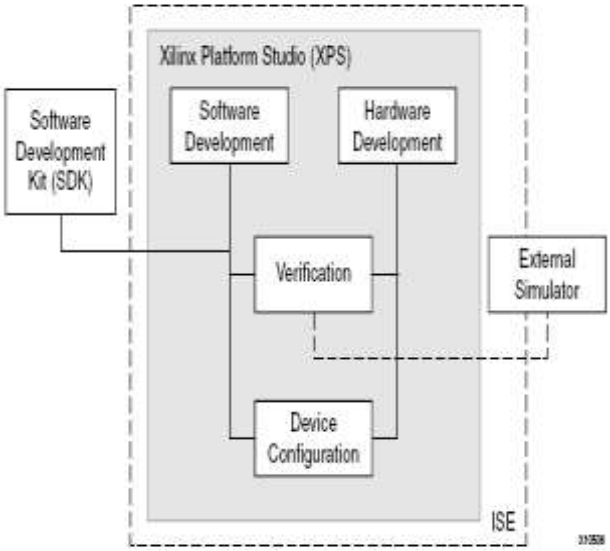

Figure 2. EDK tool flow block dig

3.4. GPIO

A range of applications can be developed with Microblaze processor. The applications may range from a simple state machine to Full custom Embedded system. DAS is a full custom Embedded System, hence it uses UART, GPIO, Timer & Interrupt controller, External memory along with Microblaze , The expansion headers provided on FPGA board can be configured as a channel of communication between GPIO (target to Microblaze) and ADC 0809. EDK 12.4 tool helps to provide the communication between the master and slave by using Base system Builder (BSB) [5]

3.5.Digital Recorders

These records the data obtained by the Microblaze in non-volatile memory for future reference. Normally UART can be used as a soft IP for recording the values. This paper implements the digital display of equivalent temperature obtained by the Transducer under supervision of Microblaze.

4. IMPLEMENTATION USING MICROBLAZE

The implementation of data acquisition system include

- Micro Blaze
- User machine interface
- Shared memory

A. Micro blaze

Micro blaze is a 32 bit RISC Harward architecture soft core processor with advanced architecture options i.e., PLB interface, memory management unit, instruction and data-side cache, configurable pipeline depth, floating point unit, etc., .it has over 70-user configurable options, enabling virtually any processor use case from a very small footprint microcontroller

to a high performance computive-intensive system running Linux.[5]

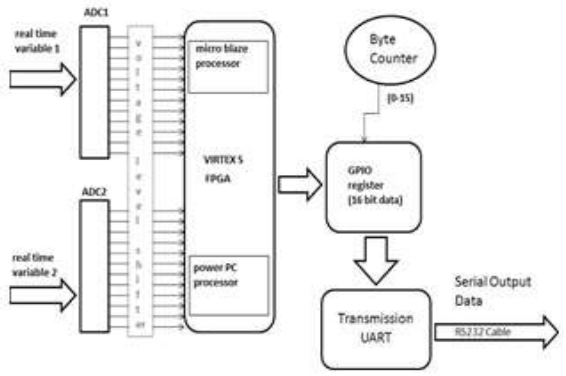

Figure 3. Flow Diagram for FPGA

B .User Machine Interface

The physical interface between FPGA and ADC is made through a GPIO peripheral IP. A timer peripheral IP is added to the project to allow the MicroBlaze to measure time delays. A delay function will be used to create the required signal timing for the ADC interface.

C. Shared Memory

A Dual Port Block Random Access Memory (DP-BRAM) is the memory used to store the values of coefficients computed by Microblaze processor. Block RAM (BRAM) IP Block and XPS BRAM Controller IP are added to the Microblaze system to implement Dual Port BRAM.

D. Process Description

The temperatures from two different analog sensors i.e., RTD are converted to digital form using ADC and the latter is fed to Microblaze Processor operating at 125 MHz through voltage translators. To make proper communication with ADC, the Microblaze processor utilizes General Purpose Input Output (GPIO) soft IP in a programmed way for sending the signals from Microblaze and vice versa. The ADC is initiated by one of the data received by the FPGA is in digital format, which is processed and the temperature is displayed on screen. Here we can observe two different parameters are measured simultaneously.

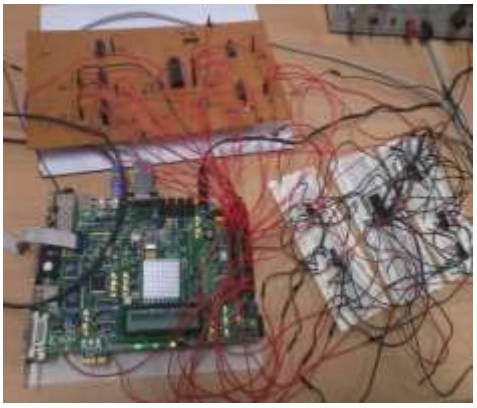

Figure 4. User machine interface

5. RESULTS

The results obtained by the Implementation of DAS using FPGA are provided in the digital format on the hexadecimal scale which can be converted into decimal format and displayed by HyperTerminal. The Table.1 gives the list of temperatures and their corresponding voltage outputs collected on HyperTerminal.

Temperature	Voltage (V)	HyperTerminal value	
		E.O.C	D7-D0
61°C	3.83	1	CC
64°C	3.88	1	D0
67°C	3.91	1	D4
70°C	3.93	1	D8
73°C	3.95	1	DC
76°C	3.96	1	E0
79°C	3.98	1	E4
82°C	3.99	1	E8
85°C	4.01	1	EC
88°C	4.21	1	F0
91°C	4.52	1	F4
94°C	4.72	1	F8
97°C	5.03	1	FE

Table.1 Temperature and Voltage output values

This data can be stored in a mass storage device like USB which can be designed as a soft IP on FPGA

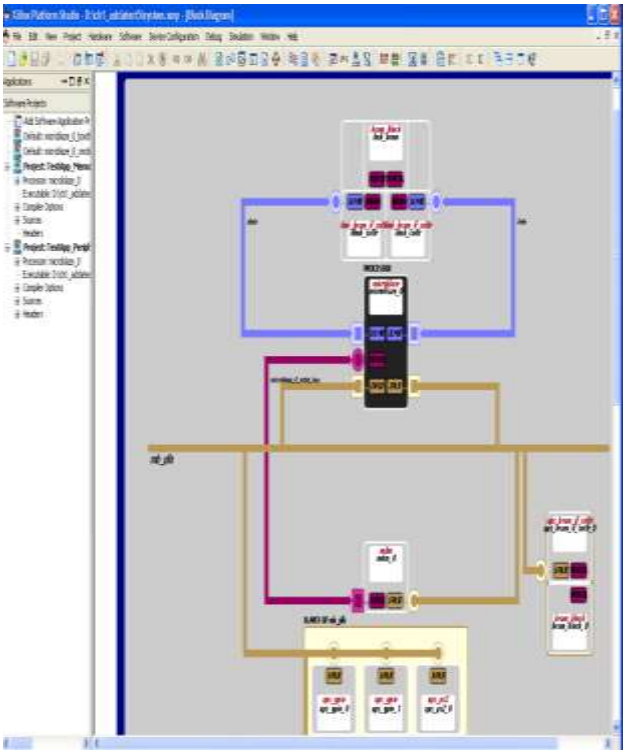

Figure 5 Block Diagram of Micro blaze Sub module

6. CONCLUSION

The Microblaze processor being a soft-core processor is added advantage compared to a hard processor which in turn is costly in terms of reconfiguring it. The programmable input output lines made it to interface with many of the real-time applications with the only cost of voltage translators or level shifters. The high speed clock of soft processor improves the speed of execution, the added advantage.

The programmable input output lines of FPGA can be used to interface more number of digital devices and the slots can also be expanded in order to increase further more number of digital devices attached. Where for the other type of processors offers limited connectivity.

7. REFERENCES

[1] *Embedded systems tool reference manual"* EDK 12.4 version.

[2] Shebli Anvar, Olivier gachelin, Pierre Kestener, Herve Le Provost, Irakli Mandjavidze,"*FPGA – based System-on-chip Designs for Real-Time Applications in Particle Physics",*14th IEEE Real-Time Conference, Stockholm, Sweden, June 6-10, 2005.

[3] S. Thanee S. Somkuarnpandit and K. Saetang," FPGA Based Multi Protocal data Acquisition System with High Speed USB Interface".

[4] "Microblaze Processor Reference Guide", Embedded Development Kit EDK 12.4 version.

[5] A. Sagahyroon,T.Al-kudairi*, "FPGA Based Acquistion of sensor Data"*,InternationalConference on Information. and Communication Technology, ,ICTTA 2006.

[6] http:/www.xilinx.com/tools/microblaze.html

[7] *"Microblaze Processor Reference Guide"*

International Journal of Science and Engineering Applications
Volume 2 Issue 7, 2013, ISSN-2319-7560 (Online)

Design and Implementation of Inverse Fast Fourier Transform for OFDM

R.Durga Bhavani
TKR College of Engineering
Hyderabad, India

D.Sudhakar
TKR College of Engineering
Hyderabad, India

Abstract: OFDM is the most promising modulation technique for most of the wireless and wired communication standards. The basic idea of OFDM is to divide the available spectrum into several sub channels, making all sub channels narrowband which experiences flat fading. OFDM uses the spectrum efficiently due to its orthogonality and prevents interference between the closely spaced carriers. OFDM provides high bandwidth efficiency because the carriers are orthogonal to each others and multiple carriers share the data among themselves. The main advantage of this transmission technique is their robustness to channel fading in wireless communication environment. The main focus of this paper is to design IFFT and FFT blocks which are used in transmitter and receiver blocks of OFDM system. The methodology used is the 8-point IFFT/FFT (DIF) with radix-2. The design unit also consists of spreader and despreader for mapping technique. The implementation is done in FPGA by using Verilog HDL. The timing simulation and synthesized results are performed and the design is analyzed by using Xilinx ISE tools.

Keywords: OFDM, IFFT, FFT, QPSK.

1. INTRODUCTION

The rapid advances in multimedia applications involve more and more transmissions of graphical data, video and audio messages. In modern communication systems, Orthogonal Frequency Division Multiplexing (OFDM) systems are used to transmit with higher data rate and avoid Inter Symbol Interference (ISI) [5].In an OFDM communication system, the broadband is partitioned into many orthogonal sub-carriers, in which data is transmitted in a parallel fashion. Thus the data rate for each sub-carrier is lowered by a factor of *N* in a system with *N* sub-carriers. By this method, the channel is divided into many narrowband flat fading sub-channels. This makes the OFDM system more resistant to the multi-path frequency selective fading than the single carrier communication system. The sub-carriers are totally independent and orthogonal to each other. The sub-carriers are placed exactly at the nulls in the modulation spectral of one another. At the peak point of one sub-carrier waveform, the sample values of other sub-carriers at the nulls are zeros and thus contribute no ISI to the sampled sub-carrier [2].

The OFDM transmitter and receiver contain Inverse Fast Fourier Transform (IFFT) and Fast Fourier Transform (FFT), respectively[6]. The IFFT/FFT algorithms are chosen due to their execution speed, flexibility and precision [3]. For real time systems the execution speed is the main concern. The IFFT block provides orthogonality between adjacent subcarriers. The orthogonality makes the signal frame relatively secure to the fading caused by natural multipath environment. As a result OFDM system has become very popular in modern telecommunication systems. The main objective of this paper is to design IFFT/FFT blocks for OFDM, because these are main blocks for modulation and demodulation in OFDM transmitter and receiver [2]. The OFDM signal is generated by implementing the Inverse Fast Fourier Transform (IFFT) at the transmitter which is used to convert frequency domain to time domain and Fast Fourier Transform (FFT) which is used to convert time domain to frequency domain at the receiver side is implemented.

The basic equation of the FFT is

$$X(k)=\sum_{n=1}^{N-1} x(n)e^{-j2\pi kn/N} \quad ,k=0,\ldots,N-1$$

On the other hand, the Inverse FFT equation is

$$x(n)=\frac{1}{N}\sum_{n=1}^{N-1} X(k)e^{-j2\pi kn/N} \quad ,n=0,\ldots,N-1$$

where N is the transform size or the number of sample points in the data frame. X(k) is the frequency output of the FFT at k^{th} point where k=0, 1, …, N-1 and x(n) is the time sample at n^{th} point with n=0, 1,…, N-1 [4].

After the brief discussion of OFDM blocks, we describe the implementation of FFT and IFFT blocks and are explained in section II that follows the OFDM Overview along with its Transmitter and Receiver blocks in section III that follows synthesis and timing simulation results using Xilinx ISE tool in section IV then Conclusion in section V and References in section VI.

2. IMPLEMENTATION OF 8-POINTS DIF IFFT.

The implementation of IFFT is simple as compared to DFT and its 8-point DIF output is derived from the input directly [1]. The figure shows the implementation can be done in three stages as shown in Fig 1. In each stage, it has four butterflies for both real and imaginary values. Each butterfly consists of upper wing and lower wing. The first stage accepts the input data directly from QPSK mapper which consists of real and imaginary values. The output of first stage is feed as the input to the second stage. The output of second stage is feed as the input to the third stage.

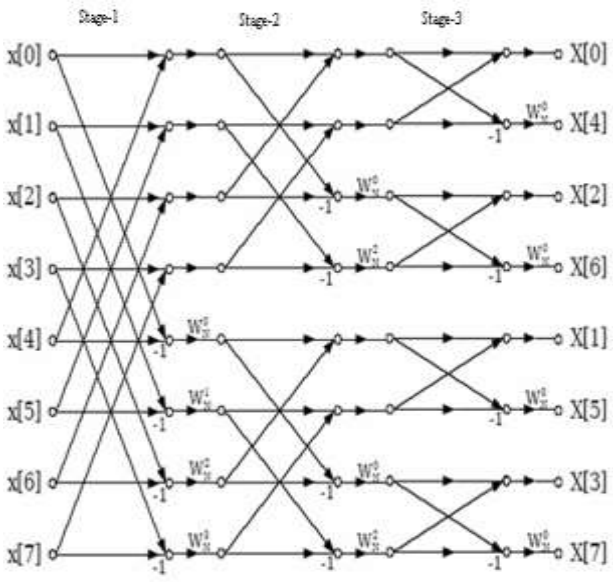

Figure 1. 8-Point IFFT(DIF)

The implementation of an 8 Point IFFT processor is done by using components like adders, subtractors, multipliers and buffers[1]. In IFFT the twiddle factor values are of unsigned values which have to be converted into binary form for the multiplication purpose.

In first stage the computation of upper wings is done by using adders and the results are stored in buffers. The computation of lower wings is done by using subtractors and multipliers. Here for every computation of lower wing there is a different twiddle factor which has to be multiplied. After the computation of both upper and lower wings the results are stored in buffers which are feed as input to the second stage.

The computation of second stage is also similar to that of first stage but the difference is that for the four lower wings there are only two twiddle factors in common. Here Again the computation results are stored in buffers and are feed to the third stage.

The computation of third stage is also similar to that of first stage but the difference is that there is only one common twiddle factor to be multiplied.

3. OFDM

Orthogonal frequency division multiplexing (OFDM) is a special case of multicarrier transmission where a single DataStream is transmitted over a number of lower rate subcarriers. In wireless communication, concept of parallel transmission of symbols is used to achieve high throughput and better transmission quality [8]. Orthogonal Frequency Division Multiplexing (OFDM) is one of the techniques for parallel transmission. The idea of OFDM is to split the total transmission bandwidth into a number of orthogonal subcarriers in order to transmit the symbols using these subcarriers in parallel.

OFDM has several advantages over single carrier modulation systems and these make it a viable alternative for CDMA in future wireless networks [8]. The advantages of OFDM are Multi path delay spread tolerance, Immunity to frequency selective fading channels, efficient modulation and demodulation, High transmission bitrates, Easy equalization, High spectral efficiency and Flexibility [6]. The disadvantage of OFDM is, it is very complex than single-carrier modulation and requires more linear power amplifier.

3.1 OFDM Transmitter

The detailed block diagram of OFDM Transmitter system is shown in the Fig 2. The components of the OFDM Transmitter are spreader, QPSK Mapper, IFFT and parallel to serial converter. The functioning of each block is explained as follows.

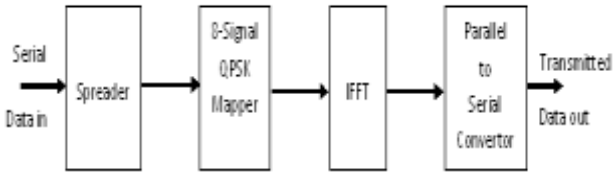

Figure 2. OFDM Transmitter

3.1.1 Spreader

The function of spreader is similar to that of serial in parallel out converter. Here the input symbols are transmitted in serial fashion and spreads out parallel in the form of pairs. These output symbols are sent to 8-signal QPSK Mapper block.

3.1.2 QPSK Mapper

The function of mapper is to convert the input data into complexed valued constellation points, according to a given constellation. Some typical constellations for wireless applications are BPSK, QPSK and QAM[7], in this work QPSK mapper is used at transmitter side. The constellation graph of QPSK can be shown in Figure 3. In QPSK there are four possible phases and therefore two bits of information conveyed within each time slot. The rate of change (baud) in this signal determines the signal bandwidth but the throughput or bit rate for QPSK is twice the baud rate.

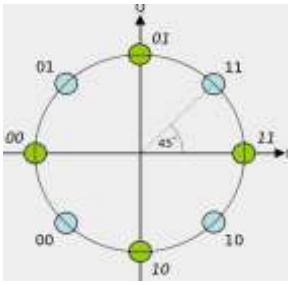

Figure 3. QPSK Constellation graph

Here in OFDM Transmitter block 8-signal QPSK Mapper is used means 8-signals each of 2-bit data is converted into constellation points and sent to the IFFT block.

3.1.3 IFFT

The IFFT transform a spectrum (amplitude and phase of each component) into a time domain signal. An IFFT converts a number of complex data points, of length that is power of 2, into the same number of points in time domain. Each data point in frequency spectrum used for an FFT or IFFT operation is called a bin. The Inverse Fast Fourier Transform (IFFT) performs N-Point IFFT operation for the received

constellation points from the QPSK Mapper.. The output is of N time domain samples. After N-point computation these values are passed through parallel to serial convertor

3.1.4 PISO

The parallel in serial out block receives the N time domain samples in parallel and transmitters out serially.

3.2 OFDM Receiver

The detailed block diagram of OFDM Receiver system is shown in the Fig 4. The components of OFDM Receiver are serial to parallel converter, FFT, demapper and despreader. The functioning of each block is explained as follows

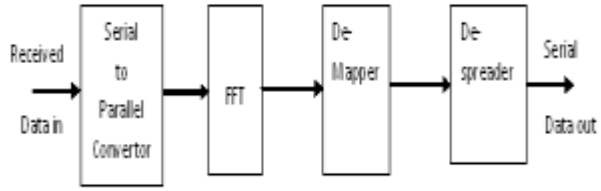

Figure 4. OFDM Receiver

3.2.1 SIPO

The received symbol from the OFDM Transmitter is in time domain which is passed serially through the serial to parallel converter and outs data in parallel. These parallel symbols are sent to Fast Fourier Transform (FFT) block.

3.2.2 FFT

FFT Converters time domain to frequency domain. The parallel symbols which are received from serial to parallel converter perform N-Point FFT operation and sends to demapper.

3.2.3 Demapper

The function of demapper is to convert complexed valued constellations points to symbols. Here in OFDM Receiver block Demapper is used to convert constellation points to 8-signal each of 2-bit data and sent to Despreader block.

3.2.4 Despreader

The function of despreader is similar to that of parallel in serial out converter. Here the input symbols are transmitted in parallel fashion and despread out in serial form.

4. RESULTS

The presented OFDM system is shown in the above is designed using Verilog Hardware Description Language and synthesized using Xilinx Project Navigator Xilinx ISE tools. Simulation results are verified by using ISE Simulator.

The RTL Schematic view of OFDM Transmitter is shown in Fig.5 and the OFDM Receiver is shown in Fig.6.

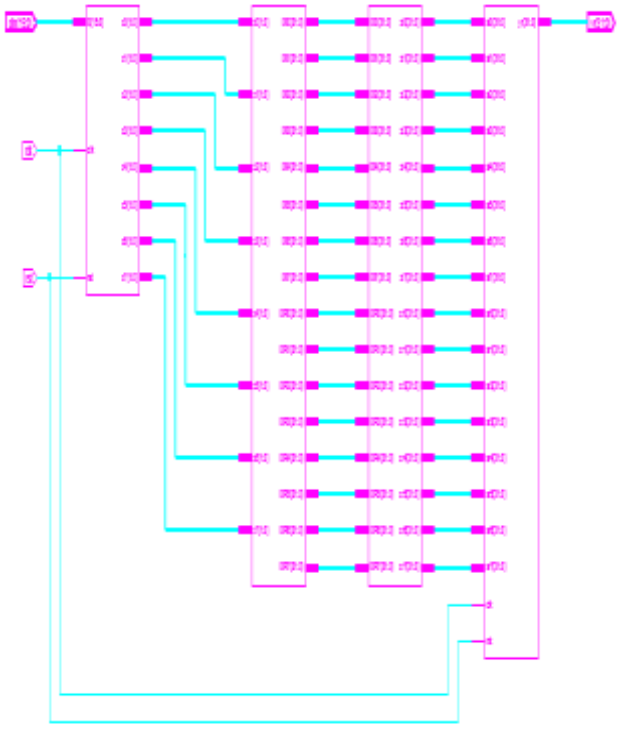

Figure 5 RTL Schematic of OFDM Transmitter

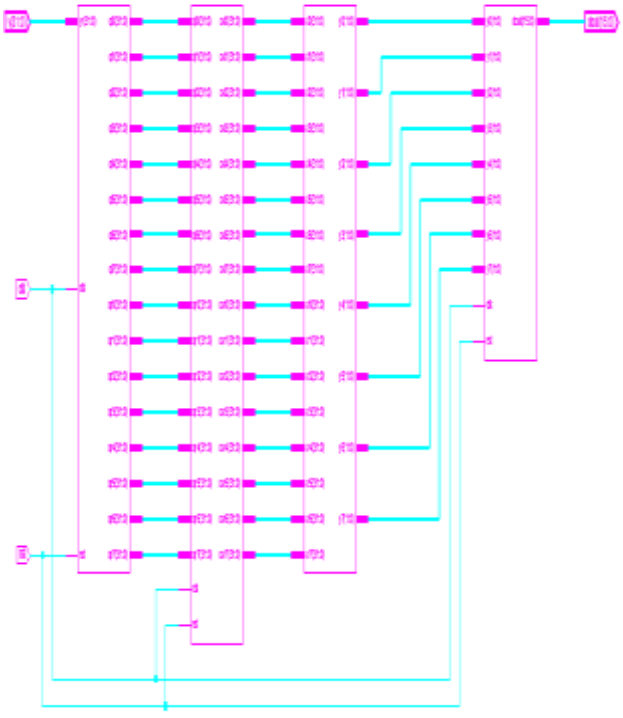

Figure 6 RTL Schematic of OFDM Receiver

Simulation result of IFFT is shown in the Fig. 7 and the result of FFT is shown in Fig.8. The result of back to back connection of OFDM Transmitter and OFDM Receiver is shown in Fig. 9.

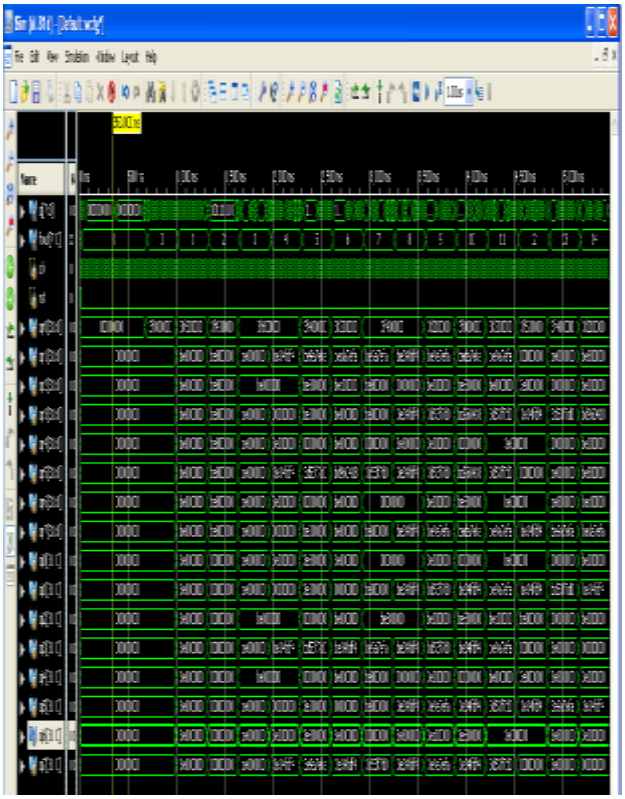

Figure 7 Simulation Result of IFFT output

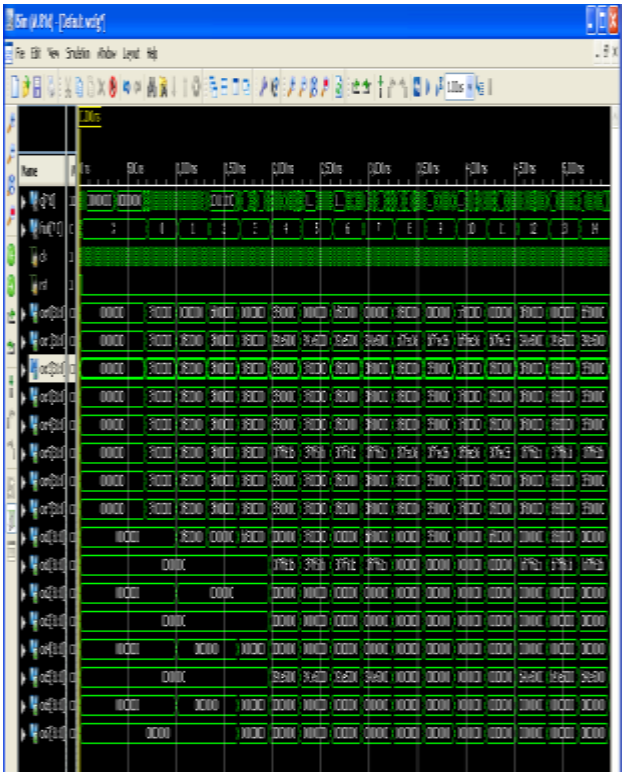

Figure 8 Simulation Result FFT output

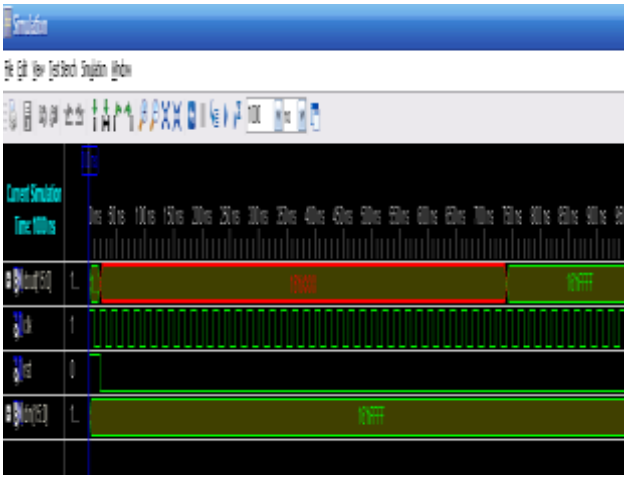

Figure 9 Simulation Result of OFDM

5.CONCLUSIONS

The design of IFFT/FFT for OFDM is implemented by using Xilinx ISE design suite. The design can be extended to implement the total OFDM transmitter and receiver with different modulation techniques like PSK, and QAM etc.

6. REFERENCES

[1] Poonam Agrawal, Vikas Gupta, "Enhanced Core Processor Blocks of OFDM System". International Journal of Electronics and Communication and Computer Engineering Volume 1 Issue 2, November 2011

[2] Manjunath Lakkannavar, Ashwini Desai, "Design and Implementation of OFDM". International Journal of Engineering and Advanced Technology(IJEAT) Volume 1 Issue 6, August 2012.

[3] K.Hari Krishna, T.Rama Rao, Vladimir A Labay, "FPGA Implementation of FFT Algorithm for IEEE 802.16e (Mobile WiMax)".International Journal of Computer Theory and Engineering, Vol. 3, No.2 , April 2011.

[4] Ahmed Saeed, M.Elbably, G.Abdelfadeel, and M.I.Eladway, "Efficient FPGA Implementation of FFT/IFFT Processor", International Journal of Circuits, Systems and Signal Processing.

[5] M.A..Mohamed, A.S.Samarah, and M.I.Fath Allah, "Implementation of the OFDM Physical Layer Using FPGA". International Journal of Computer Science Issues, Vol.9, Issue 2,No 2, March 2012.

[6] Shaminder Kaur, Rajesh Mehra, "FPGA Implementation of OFDM Transceiver using FFT Algorithm" .International Journal of Engineering Science and Technology(IJEST).

[7] Intuitive guide to Principles of communications, copyright 2002 Charan Langton, revised Dec 2005.

[8] M.A.Mohamed, A.S.Samarah and M.I.Fath Allah " Implementation of Adaptive OFDM System using FPGA". International Journal of Computer Science Issues,Vol.9, Issue 3, No 3,May 2012